

(51) International Patent Classification:

A61B 5/00 (2006.01) A61B 5/02 (2006.01)
G06F 9/44 (2006.01) A61B 5/145 (2006.01)
A61B 5/01 (2006.01)

(21) International Application Number:

PCT/US2015/037173

(22) International Filing Date:

23 June 2015 (23.06.2015)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/017,895 27 June 2014 (27.06.2014) US
62/017,908 27 June 2014 (27.06.2014) US
14/535,395 7 November 2014 (07.11.2014) US
14/535,401 7 November 2014 (07.11.2014) US

(71) Applicant: WELCH ALLYN, INC. [US/US]; 4341 State Street Road, Skaneateles Falls, New York 13153 (US).

(72) Inventors: GONDEK, Cory R.; c/o Welch Allyn, Inc., Attn: Legal Department, 4341 State Street Road, Skaneateles Falls, New York 13153 (US). NAGASUBRAMANIAN, Soundharya; c/o Welch Allyn, Inc., Attn: Legal Department, 4341 State Street Road, Skaneateles Falls, New York 13153 (US). KOVITZ, Paul S.; c/o Welch Allyn,

Inc., Attn: Legal Department, 4341 State Street Road, Skaneateles Falls, New York 13153 (US). ALFRED, Charles; c/o Welch Allyn, Inc., Attn: Legal Department, 4341 State Street Road, Skaneateles Falls, New York 13153 (US). GOLDMAN, Dan E.; c/o Welch Allyn, Inc., Attn: Legal Department, 4341 State Street Road, Skaneateles Falls, New York 13153 (US).

(74) Agents: KALINSKY, Robert A. et al.; Merchant & Gould P.C., P.O. Box 2903, Minneapolis, Minnesota 55402 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,

[Continued on next page]

(54) Title: LOW ACUITY VITALS SYSTEM ARCHITECTURE

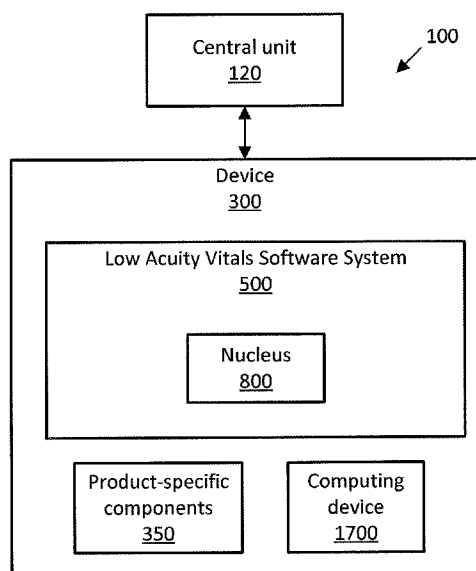


FIG. 1

(57) Abstract: A medical device including a non-transitory storage device storing a plurality of modular software components programmed to provide medical device functionality is disclosed. The software components can be configured to run independently as a daemon, to be modified without recompiling unchanged software components and to communicate to each other through a software component. A system including a central unit configured to communicate with a medical device is also disclosed. A medical device software architecture stored on a non-transitory, computer readable storage medium including a plurality of modular software components programmed to provide medical device functionality is also disclosed.



LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, **Published:**
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, — *with international search report (Art. 21(3))*
GW, KM, ML, MR, NE, SN, TD, TG).

LOW ACUITY VITALS SYSTEM ARCHITECTURE

[0001] This application is being filed on June 22, 2015, as a PCT International Patent application and claims priority to U.S. Patent Application Serial No. 62/017,895 filed on June 27, 2014, U.S. Patent Application Serial No. 62/017,908 filed on June 27, 2014, U.S. Patent Application Serial No. 14/535,401 filed November 7, 2014, and U.S. Patent Application Serial No. 14/535,395 filed on November 7, 2014, the entire disclosures of which are incorporated by reference in their entirety.

BACKGROUND

[0002] Many medical devices require software in addition to a measurement component for their operation. Devices developed and released at different times can have different software versions that may not be compatible to communicate with other devices or to a central monitoring unit. Also, a software architecture where software items cannot be reused, or where a change to one software component requires changes to all software components, can inhibit the pace of development and can increase the cost of maintaining such a system.

SUMMARY

[0003] In one aspect, a medical device including a non-transitory storage device storing a plurality of modular software components programmed to provide medical device functionality is disclosed. The software components can be configured to run independently as a daemon, to be modified without recompiling unchanged software components and to communicate to each other through a software component. The software components can also include a plurality of reusable assets where at least one of the reusable assets is configured to enable communication between the medical device and a central unit, and one of the reusable assets can include a proxy for interfacing with a measurement module. In embodiments, the measurement module can be a saturated oxygen module, a non-invasive blood pressure module or a thermometry module.

[0004] The reusable assets can be one or more of the following: a component manager configured to start up and shut down at least one of the software components; a capability registry configured to provide a bulletin board system where software components can post common object-oriented architecture messages and receive messages about changes to any of the posted messages; a configuration manager configured to maintain persistent configuration of the device and provide a device configuration to client software components; a workflow manager

configured to add, remove, start or stop manager scripts in the device; a virtual operating system configured to interface with an operating system adaptor; a virtual hardware configured to interface with a hardware adaptor; and a message manager configured to provide an inter-process communications bus for sending common object-oriented architecture messages between the software components.

[0005] In another aspect, a system including a central unit configured to communicate with a medical device is also disclosed. The central unit can include a user interface, a first communication element, and a processing element. The medical device can include a second communication element configured to communicate with the first communication element, a tangible, non-transitory storage device storing a plurality of modular software components, where each software component is configured to run independently as a daemon, where each software component is configured to be modified without recompiling unchanged software components, and where the software components are configured to communicate to teach other through a software component.

[0006] The medical monitoring system's software components can also include a plurality of reusable assets, and at least one of the plurality of reusable assets is configured to enable communication between the medical device and the central unit. The medical device can be a measurement module and one of the reusable assets can be a proxy for interfacing with a measurement module and the measurement module can be a saturated oxygen module, a non-invasive blood pressure module, or a thermometry module.

[0007] The reusable assets can be one or more of the following: a component manager configured to start up and shut down at least one of the software components; a capability registry configured to provide a bulletin board system where software components can post common object-oriented architecture messages and receive messages about changes to any of the posted messages; a configuration manager configured to maintain persistent configuration of the device and provide a device configuration to client software components; a workflow manager configured to add, remove, start or stop manager scripts in the device; a virtual operating system configured to interface with an operating system adaptor; a virtual hardware configured to interface with a hardware adaptor; and a message manager configured to provide an inter-process communications bus for sending common object-oriented architecture messages between the

software components. Each of the component manager, message manager, capability registry, configuration manager and workflow manager can be operating system and hardware agnostic.

[0008] In a further aspect, a medical device software architecture stored on a non-transitory, computer readable storage medium including a plurality of modular software components programmed to provide medical device functionality is also disclosed. The architecture can include a plurality of modular software components, where each software component is configured to run independently as a daemon, to be modified without recompiling unchanged software components, and to communicate to each other through a message manager, the message manager is configured to provide an inter-process communications bus for sending common object-oriented architecture messages between the software components. The modular software components can include reusable assets being configured to plug into the medical device software system and configured to provide a functionality. At least one of the reusable assets can be configured to enable communication between the medical device and a central unit.

[0009] The reusable assets can be one or more of the following: a component manager configured to start up and shut down at least one of the software components; a capability registry configured to provide a bulletin board system where software components can post common object-oriented architecture messages and receive messages about changes to any of the posted messages; a configuration manager configured to maintain persistent configuration of the device and provide a device configuration to client software components; a workflow manager configured to add, remove, start or stop manager scripts in the device; a virtual operating system configured to interface with an operating system adaptor; a virtual hardware configured to interface with a hardware adaptor; and a message manager configured to provide an inter-process communications bus for sending common object-oriented architecture messages between the software components. Each of the component manager, message manager, capability registry, configuration manager and workflow manager can be operating system and hardware agnostic.

DESCRIPTION OF THE FIGURES

[0010] FIG. 1 illustrates a block diagram of an example medical device.

[0011] FIG. 2 illustrates an example medical device of FIG. 1.

[0012] FIG. 3 illustrates another example medical device of FIG. 1.

[0013] FIG. 4 illustrates a block diagram illustrating an example low acuity vitals software/hardware system.

- [0014] FIG. 5 illustrates the components of an example embodiment of the nucleus.
- [0015] FIG. 6 illustrates an example environment of the low acuity vitals system.
- [0016] FIG. 7 illustrates example components of a medical device of the system of FIG. 1.

DETAILED DESCRIPTION

[0017] Medical environments, such as clinics, hospitals, nursing homes, care facilities, and at-home care stations, can rely on peripheral devices that communicate, display, and coordinate the monitoring and maintenance of a patient's health. Having the software components of a particular model and make of a medical monitoring device associated with a monitoring system, such as a specific SpO₂ monitor, can limit the flexibility of the system with respect to future improvements or modifications. Moreover, it can be disadvantageous to use a fixed software architecture that requires propagating any change, minor or major, to every software component and service within the system. The instant disclosure is directed to a solution to the above-described limitations of current medical software systems.

[0018] As used herein, "medical device" refers to any device that can be used in a medical environment. Examples of medical devices include, but are not limited to, thermometers, blood pressure sensors, heart rate sensors, pulse rate sensors, SpO₂ sensors, devices to monitor vital signs, devices for metering fluids, etc.

[0019] A first component is the low acuity vitals system architecture. The system architecture incorporates a compartmentalized design wherein each process can be run as a daemon and run separately. Thus, each process in the system is separate. Advantages of the system include, for example, that, unlike existing systems, if a change is made to the code of a process, the coding for each process does not need to be similarly adjusted. For example, if the code associated with a temperature process is changed, that change to the code does not need to be propagated to every other process in the system.

[0020] A second component is the nucleus architecture comprising separate applications. The nucleus architecture can be used to communicate information between separate software services which separates the interfaces of the services from the internal operation of the services. This structure can enable changes to individual services to be made without affecting the safety or operation of the other services. Additionally, the nucleus can also provide a base of functionality for saving data through device power cycles, starting services and monitoring the started services for functionality, providing a service where data can be stored temporarily such

that other services can be restarted and retrieve the system state, and a service to provide dynamic scripted workflows to quickly respond to customer needs. The nucleus can enable communication between separate software services, which separates the interfaces of the services from the internal operation of the services. In a preferred embodiment, the nucleus is agnostic to the particular operating system.

[0021] An example medical product system 100 is shown in FIG. 1. The example medical product system 100 includes an optional central unit 120, a device 300 comprising a computing device 1700 that includes product-specific software components 350, and a low acuity vitals software system 500, which includes a nucleus 800. Other embodiments may include more or fewer components.

[0022] In some embodiments, the device 300 can communicate via a wired or wireless protocol with the central unit 120. In some embodiments, the device 300 is the central unit 120. The central unit 120 can include a computing device, for example the computing device 1700 shown and described in detail with reference to FIG. 7, and a user interface which can include a display and input capability. In embodiments, the central unit 120 can display and/or control the example medical devices described herein.

[0023] In embodiments, the device 300 can be an external component, device, service, or service tool. The device 300 can include other product-specific components 350, such as sensor components in a thermometer, not shown. Additional examples and discussion of types of devices 300 are described in more detail with reference to peripherals 600, hardware 610 and services 620, below.

[0024] The low acuity vitals software system 500 (LAV architecture) is a software architecture used by the device 300. The low acuity vitals software system 500 includes a nucleus 800. The LAV architecture 500 is shown and described in more detail with reference to FIG. 4 below.

[0025] Example device 300 also includes a computing device 1700 for processing and communication. The computing device 1700 includes product-specific software components 350. The computing device 1700 is shown and described in more detail below with reference to FIG. 7.

[0026] Figure 2 illustrates one example of the medical device 104. The medical device 104 is shown on a mobile cart, and the medical device 104 is programmed to provide the functionalities

described herein. The medical device 104 includes a display screen 218 and includes the ability to execute multiple workflows or profiles. In some embodiments, the display screen 218 is a touch screen. Other embodiments can include more or fewer components than those shown in FIG. 2, or include different components that accomplish the same or a similar function.

[0027] The medical device 104 is able to operate within one or more profiles. A profile is a series of one or more tasks that a user of the medical device 104 performs. When the medical device 104 operates within a profile, the medical device 104 provides functionality suitable for assisting the user in performing the profile. When the medical device 104 operates within different profiles, the medical device 104 provides different functionality.

[0028] When the medical device 104 is manufactured, the medical device 104 is configured to be able to operate within one or more profiles. After the medical device 104 is manufactured, the medical device 104 can be reconfigured to operate within one or more additional profiles. In this way, a user can adapt the medical device 104 for use in different profiles as needed.

[0029] In various embodiments, the medical device 104 operates within various profiles. For example, in some embodiments, the medical device 104 can operate within an interval profile or a non-interval profile. Example types of non-interval profiles include, but are not limited to, a spot check profile, an office profile, and an interval profile. An example of an interval profile includes, but is not limited to, an intervals profile.

[0030] In example embodiments, the names for the profiles can be defined by the user. For example, the user can rename an “interval profile” as “ED 3 North” or any other nomenclature as desired to provide more context to the user.

[0031] When the medical device 104 is operating within the interval profile, the medical device 104 obtains a series of measurements of one or more physiological parameters of a single monitored patient over a period of time. In addition, the medical device 104 displays, on the display screen 218, an interval profile home screen. The interval profile home screen contains a representation of a physiological parameter of the monitored patient. The representation is based on at least one measurement in the series of measurements. A representation of a physiological parameter is a visible image conveying information about the physiological parameter.

[0032] For example, when the medical device 104 is operating within the interval profile, the medical device 104 can obtain a blood pressure measurement of a single patient once every ten minutes for six hours. In this example, the medical device 104 displays an interval profile home

screen that contains a representation of the patient's blood pressure based on a most recent one of the blood pressure measurements. In this way, a user of the medical device 104 can monitor the status of the patient.

[0033] When the medical device 104 is operating within a non-interval profile, the medical device 104 obtains a measurement of one or more physiological parameters from each patient in a series of patients. In addition, the medical device 104 displays a non-interval profile home screen on the display screen 218. The non-interval profile home screen contains a representation of the physiological parameter of a given patient in the series of patients. The representation is based on the measurement of the physiological parameter of the given patient.

[0034] In one example, when the medical device 104 is operating within a spot check profile, the medical device 104 obtains blood pressure measurements from a series of previously-identified patients. In this other example, the medical device 104 displays a spot check profile home screen containing a blood pressure measurement of a given patient in the series of previously-identified patients. In this way, a user of the medical device 104 can perform spot checks on the blood pressures of patients who have already been admitted to a hospital.

[0035] As used in this document, a patient is a previously identified patient when the medical device 104 stores information regarding the identity of the patient. In another example, when the medical device 104 is operating within an interval profile, the medical device 104 can obtain a single blood pressure measurement from each patient in a series of unidentified patients as the patients arrive at a hospital.

[0036] The interval profile home screen may be different than the non-interval profile home screen. Further, as discussed below, the navigation options associated with the different profiles allows for efficient monitoring based on the environment in which the device is used. In various embodiments, the interval profile home screen is different than the non-interval profile home screen in various ways. For example, in some embodiments, the interval profile home screen includes at least one user-selectable control that is not included in the non-interval profile home screen. In other embodiments, a representation of a physiological parameter in the interval profile home screen has a different size than a representation of the same physiological parameter in the non-interval profile home screen.

[0037] An additional example of the medical device 105 is shown in Figure 3. In this example, the medical device 105 is similar to that of the medical device 104 described above. In

embodiments, the medical device 105 is mounted on a wall. The medical device 104 is programmed in a manner similar to that described above to monitor physiological parameters of a patient. In some embodiments, the medical device 105 is a stand-alone device, which can mean that is not part of a mobile cart and it is not part of a wall-mounted station.

[0038] In the examples described herein, the medical devices 104, 105, 106 are computing devices that have been programmed to perform special, complex functions. These specially-programmed devices function to manipulate and provide data to the users in an improved form factor and with greater efficiency.

[0039] For example, as described further below, the medical devices 104, 105, 106 are specially programmed to provide the user with an improved interface during initial use of the devices. This allows the user to more efficiently select a profile for controlling the functionality of the device.

[0040] In addition, the medical devices 104, 105, 106 are specially programmed to assist the users once vital signs information is captured from the patients. For example, the devices are programmed to more efficiently and easily capture additional contextual information that is needed when saving vital signs data to a permanent record, such as an EMR record. This is accomplished using an interface that is more intuitive and robust.

[0041] The medical devices 104 and 105 shown in Figures 2 and 3 are only examples of a medical device. In some examples described herein, the medical devices 104 and 105 are portable devices. In other examples, the medical devices 104 and 105 are non-portable devices, such as computing devices like workstations. All different types of medical devices used to collect patient data can be used. Many configurations are possible.

[0042] FIG. 4 is a block diagram illustrating an example low acuity vitals (LAV) software/hardware system 501. The example system 501 includes the software components of an example low acuity vitals software system 500. Also shown are example components that can interact with the LAV system 500, such as peripherals 600, hardware 610 and services 620. The LAV system includes reusable assets 502, comprising reusable components 504 and reusable capabilities 506, a base Linux system 510, applications 508 and Nucleus 800. Other embodiments may include more or fewer components.

[0043] Example peripherals 600 can include a display, such as an LCD display, a saturation of peripheral oxygen (SpO₂) module, a non-invasive blood pressure (NIBP) module, a

SureTemp thermometry module, a Braun thermometry module, a printer, an accessory power management stand that can contain extra battery power, USB ports and work surface, a barcode reader, a nurse call, a Newmar radio, a weight scale, a touch screen, a speaker, and a buzzer.

[0044] Example hardware 610 that can interact with the LAV system 500 can include a Domain Name System (DNS) server to acquire service IPs, a Dynamic Host Configuration Protocol (DHCP) server to acquire IP addresses, a Network Time Protocol (NTP) server to acquire coordinated universal time (UTC), and an NRS service to acquire a service IPs.

[0045] Example services 620 can acquire software error logs, event logs and device configuration. The example services 620 can include a service monitor and a service tool. In embodiments, the service monitor or service tool can be a software application that can interface with the peripheral 600 or hardware 610 to perform software upgrades and download service information from the device.

[0046] Reusable assets 502 can include reusable components 504 with reusable capabilities 506. Generally, a reusable capability 506 can be an abstract interface definition that describes the functionality a software item will provide to the software system. Generally, a reusable component 504 can be a software item designed to plug into a software system and provide a set of functionality. Reusable components 504 can implement a reusable capability 506 by conforming to a specific interface.

[0047] Examples of the reusable assets 502 include an SpO2 proxy for interfacing with an SpO2 module, an NIBP proxy for interfacing with a NIBP module, a SureTemp proxy for interfacing with a SureTemp thermometry module, a Braun Temperature proxy for communicating with a Braun thermometry module, an audio manager including a set of sound files and that can control the audio hardware for a speaker and/or a buzzer, a Newmar Manager for configuring, controlling and monitoring information about a Newmar radio, an heart rate/pulse rate derived channel that can specify a channel of common object-oriented architecture (COOA) messages that are derived from the output of either or both the NIBP or SpO2 modules, and an off-the-shelf board support that includes drivers for standard and LAV-specialized hardware, which can include the Hardware Adapter that implements the virtual hardware 814 API shown and described in more detail in FIG. 5 below, EEPROM, USB client, Ethernet, touch screen and a light sensor. As used herein, COOA is a software architecture that can incorporate

an object-oriented data model with a publish/subscribe bus architecture, and it can be used to communicate information between software items.

[0048] Example base Linux system 510 components can include off-the-shelf software products such as Sqlite3, syslog-ng, package manager, QT libraries, NTP client, Python interpreter, U-Boot, Linux kernel and MLO. Additional components of the example base Linux system 510 are possible.

[0049] The example nucleus 800 is a set of portable and reusable software items designed to provide a base set of functionality for all, or substantially all, embedded medical devices that use the LAV architecture 500. The example nucleus is shown and described in more detail with reference to FIG. 5 below.

[0050] Applications 508 are program modules 1724 that are stored in a mass storage device 1714. Example applications 508 in the LAV software system 500 can include a power manager, a connectivity manager, a patient data manager, a user interface, a barcode manager, an alarms manager, a weight app manager, a log manager, a host proxy, a configuration application, a scale handler, a time manager, a system controller, a debug interface, a service monitor proxy, a software update component, a Bluetooth™ manager, a printer manager, and a peripheral interface controller (PIC) proxy.

[0051] Some of the applications 508 can interface with hardware or external components. For example, the power manager can communicate with the power control of the computing device 1700; the barcode manager can interface with a supported barcode reader to receive and publish data and control the power state of the barcode reading; the alarms manager can receive all alarm notifications and interface with the nurse call hardware; the scale handler can interface with an external weight scale; the Bluetooth™ manager can control the interface to the Bluetooth™ module; the user interface module can communicate with an external display, such as a touch-sensitive LCD display; the printer manager can interface with a printer; and the PIC proxy can interface with a PIC processor to receive battery and charge information and set LED status on the optional accessory power management stand (APM).

[0052] FIG. 5 is a block diagram of the components in an example embodiment of the nucleus 800. The example embodiment 800 includes a component manager 802, a message manager 804, a capability registry 806, a configuration manager 808, a workflow manager 810, a virtual operating system 812 that interfaces with an operating system adaptor 816, and a virtual

hardware 814 component that interfaces with a hardware adapter 818. The nucleus 800 is a cross-product, portable framework that can support embedded medical device development. Other embodiments may include more or fewer components of the nucleus 800.

[0053] In some embodiments, the component manager 802 can start up the software system including the nucleus 800 and LAV 500 software items, except the Linux kernel, the board support package and the software items started up by the init scripts or before the init program is executed. The example component manager 802 can also shut down the system and perform process monitoring for stuck or failed software items that have registered for process monitoring with the component manager 802.

[0054] In embodiments, the component manager 802 can start up the other nucleus 800 software components. Then the component manager 802 can invoke an external application or script to start up the product-specific software components and take any other actions to bring the product to a normal run state. Also, the example component manager 802 can provide a set of COOA messages that can be used by software components to interact with the component manager 802 as well as a class that can be used by a client to simplify usage of the component manager 802 functionality.

[0055] In some embodiments, the message manager 804 can provide an inter-process communications bus for sending COOA messages between software items that communicate using the example nucleus 800 architecture. The software items that can communicate through the message manager 804 can include software items from within the nucleus 800 and those outside the nucleus, provided they can communicate using COOA messages.

[0056] In embodiments, the example message manager 804 can support publish/subscribe and directed send messages. Also, in embodiments, the message manager 804 can provide a set of messages to subscribe and unsubscribe for messages as well as a class that can be used by software components to simplify the creation, sending and receiving of messages to and through the message manager 804.

[0057] In some embodiments, the capability registry 806 can be a bulletin board system where software items can post COOA messages and receive changes to any information posted. The messages can include information about connected hardware, manufacturer of the hardware, software version, and anything else that could be needed by other software components to make use of the available software and hardware capabilities. The messages can also have unique

identifiers. The software items that can use the capability registry 806 for posting messages can include all software items designed to communicate using COOA messages and the software items in the nucleus 800.

[0058] In embodiments, the capability information can be posted as COOA-format messages and the information can persist until the system is shut down, the posting software component shuts down, or until another COOA message with the same COOA ID is posted. In some embodiments, a COOA message with the same COOA ID will replace an existing COOA message.

[0059] Software components in the example LAV system 500 can subscribe through the message manager 804 for COOA messages which are posted to the capability registry 806. In turn the capability registry 806 can publish the COOA messages when they are changed or added. In embodiments, an event message is published when a capability is removed and a “SendTo” is used to notify a component when a non-existent capability COOA message is requested.

[0060] In embodiments, software components can send a request message to the capability registry 806 when first subscribing for information in the capability registry 806 to get the current state of the COOA messages. The capability registry 806 can use a directed send to send the requested values back to the requestor, which can prevent possible race conditions between requesting components and updating components.

[0061] In embodiments, the example configuration manager 808 can maintain persistent configuration of the device and can provide this information to client software components. The configuration manager 808 can persist configuration through power cycles. Also, the configuration manager 808 can manage multiple versions of each configuration COOA message and can return different versions of a configuration message depending upon the requested item. The COOA message version number can be used to support a versioning scheme of the configuration messages so that the software system can be upgraded and downgraded and still be able to receive and understand the configuration messages.

[0062] If, in embodiments, the upgraded software then upgrades a configuration message to a new, higher version, the old configuration message can still be available if the software is downgraded. If a change is then made to the upgraded configuration, the configuration manager 808 can opt to not synchronize changes between versions of configuration messages. If the

software is downgraded, the configuration manager 808 can use the old values that were in place before the software upgrade even if the subsequent changes are compatible with the old software.

[0063] The configuration manager 808 can use product-provided plugins to interface to different types of storage devices. An example is a plug-in that stores data in a regular file for non-permanent configuration and another plug-in that saves data to an EEPROM for permanent configuration and licenses.

[0064] Software components that want to receive configuration data can subscribe to the COOA IDs through the message manager 804 as with other data. In embodiments, after a software component has subscribed to configuration messages, it can send a request to the configuration manager 808 to receive the current values. To prevent a potential race condition, software components that own some configuration, in some embodiments, do not publish changes through the message manager 804, but use the store functionality of the configuration manager 808 and allow the configuration manager 808 to publish the change.

[0065] In some embodiments, the workflow manager 810 can run programs, including, but not limited to, scripts, to provide dynamically changing functionality to the device. The example workflow manager 810 can add, remove, start and stop programs, including manager scripts, on demand, as well as provide information about managed programs and/or scripts.

[0066] In some embodiments, the virtual operating system 812 can be a wrapper layer to provide a fixed application programming interface (API) between software components and the underlying operating system. In embodiments, the operating system adapter 816 can include software that can cause function calls made using the virtual operating system 812 API to result in corresponding function calls in the operating system.

[0067] In some embodiments, the virtual hardware 814 can provide a fixed API between software components and the underlying hardware and device drivers. In embodiments, the hardware adapter 818 can include software that can cause function calls made using the virtual operating system 812 API to result in corresponding function calls to the operating system that accesses the underlying hardware. The virtual hardware 814 can enable abstraction of differences across operating system and hardware instances for these functions such that reusable components do not require re-write to access those components.

[0068] FIG. 6 is a block diagram of an example LAV system environment 701. Included in the example environment 701 is the LAV system 501, which can include services 620 and peripherals 600, stakeholders 702, and a service monitor 704.

[0069] Stakeholders 702 include the groups or personnel on the service and customer sides that can interact with the LAV system 501. Example stakeholders 702 can include acquirers, assessors, communicators, developers, maintainers, production engineers, suppliers, support staff, system administrators, testers and users.

[0070] The acquirers can include customers who oversee the procurement of the system or product. The assessors can include groups or personnel that oversee the system's conformance to standards and legal regulation. The communicators can explain the LAV system 501 to other stakeholders through documentation and training materials. The developers can construct and deploy the system from specifications. Developers can also be concerned with build standards, platform, language, tools, maintainability, flexibility, and the preservation of knowledge over time.

[0071] The maintainers can manage the evolution of the LAV system 501 and can be concerned with system documentation, instrumentation, diagnostics, production change control and the preservation of knowledge over time.

[0072] The production engineers can design, deploy and manage the hardware and software environments in which the system can be tested and run, and in which modifications to the system can be built. The suppliers can build and/or supply the hardware, software or infrastructure on which the system runs. The support staff can provide support to users of the system and can be concerned with operational and support issues, such as diagnostics and alerts.

[0073] The system administrators can run the system and can be concerned with the system 501 operation including system monitoring, management, disaster recovery, availability, resilience and scalability. The testers can test future builds to ensure they are suitable for use by evaluating the system requirements using tests to prove the requirements have been met. Users can define the system's functionality and can be the people who use the system on a regular basis. Users can be concerned with how the system functions, system performance, operations, security and support. Example users can include healthcare personnel and patients.

[0074] In embodiments, personnel and business groups can take on multiple roles, such as service personnel acting as software updater, calibrator, or tester as needed and depending on the

actions required. Additional examples of interactions with the LAV system 501 are provided below.

[0075] The example LAV system 501 can acquire vital signs from a patient. For example, the patient can have blood pressure measured with a NIBP device, temperature measured using, for example, a SureTemp or Braun thermometer, SpO2 measured, pulse rate measured from, for example, the NIBP device or SpO2, weight measured through USB or wireless-connected weight scales, and have a patient ID barcode scanned.

[0076] In embodiments, a clinician can interact with the LAV system 501 by logging in, authenticating, perform readings, reviewing readings, sending readings and using the nurse call. A clinician can log into a device using name and password, or barcode scan with or without a password. Additionally, a clinician can enter configurable modifiers and/or configurable parameters through a user interface, review historical patient vital signs records, and send selected vital signs records to a networked host system, for instance, electronic medical records or Connex CS.

[0077] A configurator can, for example, configure the LAV system 501 using a USB drive containing a configuration file or via a wired or wireless network. Additionally, a configurator can clone a LAV system 501 configuration to a storage device, such as a USB drive, and use that to configure other LAV system 501 devices. A configurator can configure the LAV system 501 through a user interface, which can be patient specific, such as alarm limits, or device settings, such as display brightness.

[0078] A calibrator can calibrate devices that use the LAV system 501, such as the NIBP module and weight scale module.

[0079] A software developer can interact with the LAV system 501 to develop and/or debug software. This can include writing and maintaining device software and gathering information to improve software performance.

[0080] A tester can also interact with the LAV system 501. The example tester can prepare test plans, create test tools, execute test plans, review test results, perform unit and integration testing, and test network connectivity, for example, using a ping or traceroute.

[0081] FIG. 7 illustrates example physical components of a computing device, such as the medical devices 102, 104. As illustrated, the device includes at least one central processing unit (“CPU”) 1708, a system memory 1712, and a system bus 1710 that couples the system memory

1712 to the CPU 1708. The system memory 1712 includes a random access memory (“RAM”) 1718 and a read-only memory (“ROM”) 1720. A basic input/output system containing the basic routines that help to transfer information between elements within the device, such as during startup, is stored in the ROM 1720. The device further includes a mass storage device 1714. The mass storage device 1714 is able to store software instructions and data.

[0082] The mass storage device 1714 is connected to the CPU 1708 through a mass storage controller (not shown) connected to the bus 1710. The mass storage device 1714 and its associated computer-readable data storage media provide non-volatile, non-transitory storage for the device. Although the description of computer-readable data storage media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, it should be appreciated by those skilled in the art that computer-readable data storage media can be any available non-transitory, physical device or article of manufacture from which the device can read data and/or instructions.

[0083] Computer-readable data storage media include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable software instructions, data structures, program modules or other data. Example types of computer-readable data storage media include, but are not limited to, RAM, ROM, EPROM, EEPROM, flash memory or other solid state memory technology, CD-ROMs, digital versatile discs (“DVDs”), other optical storage media, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the device.

[0084] According to various embodiments of the invention, the device may operate in a networked environment using logical connections to remote network devices through the network 108, such as a local network, the Internet, or another type of network. The device connects to the network 108 through a network interface unit 1716 connected to the bus 1710. The network interface unit 1716 may also be utilized to connect to other types of networks and remote computing systems. The device also includes an input/output controller 1722 for receiving and processing input from a number of other devices, including a keyboard, a mouse, a touch user interface display screen, or another type of input device. Similarly, the input/output controller 1722 may provide output to a touch user interface display screen, a printer, or other type of output device.

[0085] As mentioned above, the mass storage device 1714 and the RAM 1718 of the device can store software instructions and data. The software instructions include an operating system 1732 suitable for controlling the operation of the device. The mass storage device 1714 and/or the RAM 1718 also store software instructions, that when executed by the CPU 1708, cause the device to provide the functionality of the device discussed in this document. For example, the mass storage device 1714 and/or the RAM 1718 can store software instructions that, when executed by the CPU 1708, cause the physiological monitor device to display the home screen and other screens.

[0086] Although the example medical devices described herein are devices used to monitor patients, other types of medical devices can also be used. For example, the different components of the CONNEX™ system, such as the intermediary servers that communicate with the monitoring devices, can also require maintenance in the form of firmware and software updates. These intermediary servers can be managed by the systems and methods described herein to update the maintenance requirements of the servers.

[0087] Embodiments of the present invention may be utilized in various distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network in a distributed computing environment.

[0088] The block diagrams depicted herein are just examples. There may be many variations to these diagrams described therein without departing from the spirit of the disclosure. For instance, components may be added, deleted or modified.

[0089] While embodiments have been described, it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements can be made.

CLAIMS

What is claimed is:

1. A medical device, comprising:
a tangible, non-transitory storage device storing a plurality of modular software components programmed to provide medical device functionality;
wherein each software component is configured to run independently as a daemon;
wherein each software component is configured to be modified without recompiling unchanged software components; and
wherein the software components are configured to communicate to each other through a software component.
2. The medical device of claim 1, wherein the software components further comprise a plurality of reusable assets; and
wherein at least one of the plurality of reusable assets is configured to enable communication between the medical device and a central unit.
3. The medical device of claim 2, wherein the at least one of the plurality of reusable assets is a proxy for interfacing with a measurement module.
4. The medical device of claim 3, wherein the measurement module is a saturated oxygen module, a non-invasive blood pressure module, or a thermometry module.
5. The medical device of claim 4, wherein one of the plurality of reusable assets is a component manager configured to start up and shut down at least one of the software components;
wherein one of the plurality of reusable assets is a capability registry configured to provide a bulletin board system where software components can post common object-oriented architecture messages and receive messages about changes to any of the posted messages;

wherein one of the plurality of reusable assets is a configuration manager configured to maintain persistent configuration of the device and provide a device configuration to client software components; and

wherein one of the plurality of reusable assets is a workflow manager configured to add, remove, start or stop manager scripts in the device.

6. The medical device of claim 1, wherein the software components further comprise a plurality of reusable assets;

wherein one of the plurality of reusable assets is a virtual operating system configured to interface with an operating system adaptor; and

wherein one of the plurality of reusable assets is a virtual hardware configured to interface with a hardware adaptor.

7. The medical device of claim 6, wherein one of the plurality of reusable assets is a message manager, the message manager is configured to provide an inter-process communications bus for sending common object-oriented architecture messages between the software components.

8. The medical device of claim 7, wherein one of the plurality of reusable assets is a component manager configured to start up and shut down at least one of the software components;

wherein one of the plurality of reusable assets is a capability registry configured to provide a bulletin board system where software components can post common object-oriented architecture messages and receive messages about changes to any of the posted messages;

wherein one of the plurality of reusable assets is a configuration manager configured to maintain persistent configuration of the device and provide the configuration to client software components; and

wherein one of the plurality of reusable assets is a workflow manager configured to add, remove, start or stop manager scripts in the device.

9. A medical monitoring system, comprising:
a central unit configured to communicate with at least one medical device, comprising:
a user interface;
a first communication element;
a processing element;
the medical device comprising:
a second communication element configured to communicate with the first communication element;
a tangible, non-transitory storage device storing a plurality of modular software components;
wherein each software component is configured to run independently as a daemon;
wherein each software component is configured to be modified without recompiling unchanged software components; and
wherein the software components are configured to communicate to each other through a software component.
10. The medical monitoring system of claim 9, wherein the software components further comprise a plurality of reusable assets; and
wherein at least one of the plurality of reusable assets is configured to enable communication between the medical device and the central unit.
11. The medical monitoring system of claim 10, wherein the medical device is a measurement module;
wherein the at least one of the plurality of reusable assets is a proxy for interfacing with a measurement module; and
wherein the measurement module is a saturated oxygen module, a non-invasive blood pressure module, or a thermometry module.

12. The medical monitoring system of claim 11, wherein one of the plurality of reusable assets is a virtual operating system configured to interface with an operating system adaptor;

wherein one of the plurality of reusable assets is a virtual hardware configured to interface with a hardware adaptor;

wherein the virtual operating system is operating-system agnostic; and

wherein the virtual hardware is operating-system agnostic.

13. The medical monitoring system of claim 12, wherein one of the plurality of reusable assets is a capability registry configured to provide a bulletin board system where software

components can post common object-oriented architecture messages and receive messages about changes to any of the posted messages.

14. The medical monitoring system of claim 13, wherein one of the plurality of reusable assets is a message manager, the message manager is configured to provide an inter-process communications bus for sending common object-oriented architecture messages between the software components;

wherein one of the plurality of reusable assets is a component manager configured to start up and shut down at least one of the software components;

wherein one of the plurality of reusable assets is a configuration manager configured to maintain persistent configuration of the device and provide a device configuration to client software components; and

wherein one of the plurality of reusable assets is a workflow manager configured to add, remove, start or stop manager scripts in the device.

15. The medical monitoring system of claim 14, wherein each of the component manager, message manager, capability registry, configuration manager and workflow manager is operating system agnostic.

16. A medical device software architecture stored on a non-transitory, computer-readable storage medium, comprising:

a plurality of modular software components;

wherein each software component is configured to run independently as a daemon;

wherein each software component is configured to be modified without recompiling unchanged software components;

wherein the software components are configured to communicate to each other through a message manager, the message manager is configured to provide an inter-process communications bus for sending common object-oriented architecture messages between the software components.

17. The medical device software system of claim 16, wherein the modular software components include reusable assets, the reusable assets being configured to plug into the medical device software system and configured to provide a functionality.

18. The medical device software system of claim 16, wherein the software components further comprise a plurality of reusable assets;

wherein the message manager is one of the plurality of reusable assets;

wherein one of the plurality of reusable assets is a virtual operating system configured to interface with an operating system adaptor; and

wherein one of the plurality of reusable assets is a virtual hardware configured to interface with a hardware adaptor.

19. The medical device software system of claim 18, wherein one of the plurality of reusable assets is a component manager configured to start up and shut down at least one of the software components;

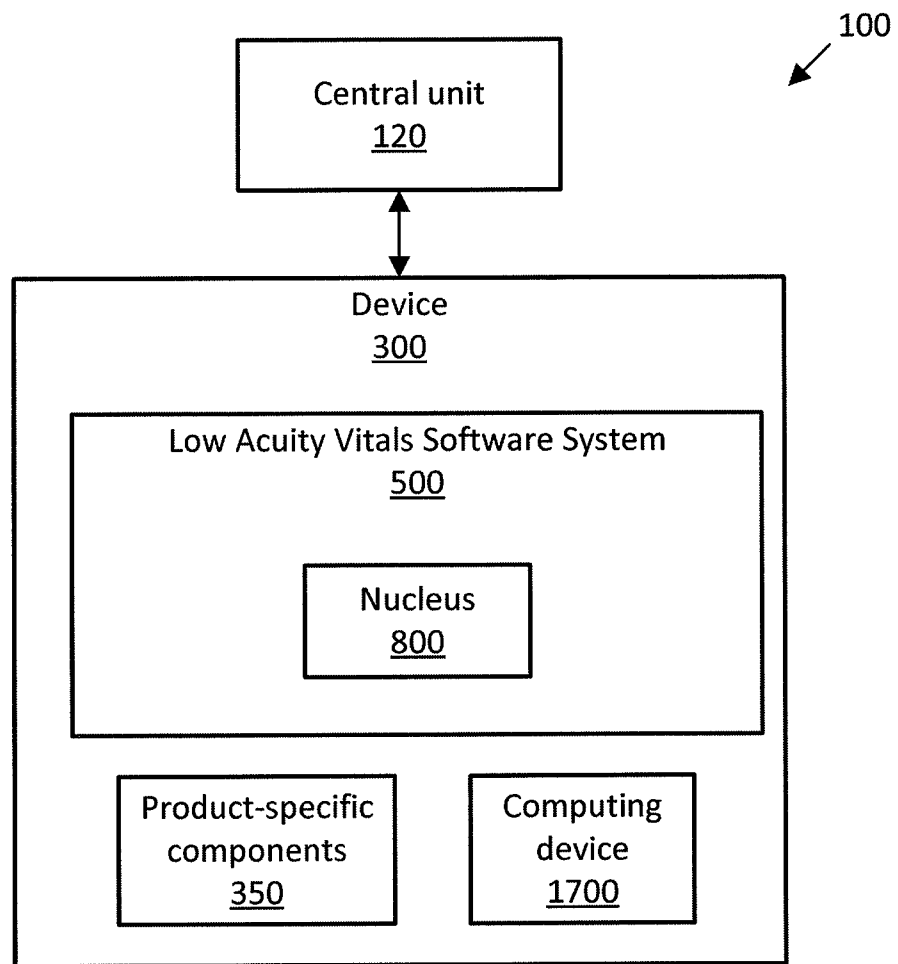
wherein one of the plurality of reusable assets is a capability registry configured to provide a bulletin board system where software components can post common object-oriented architecture messages and receive messages about changes to any of the posted messages;

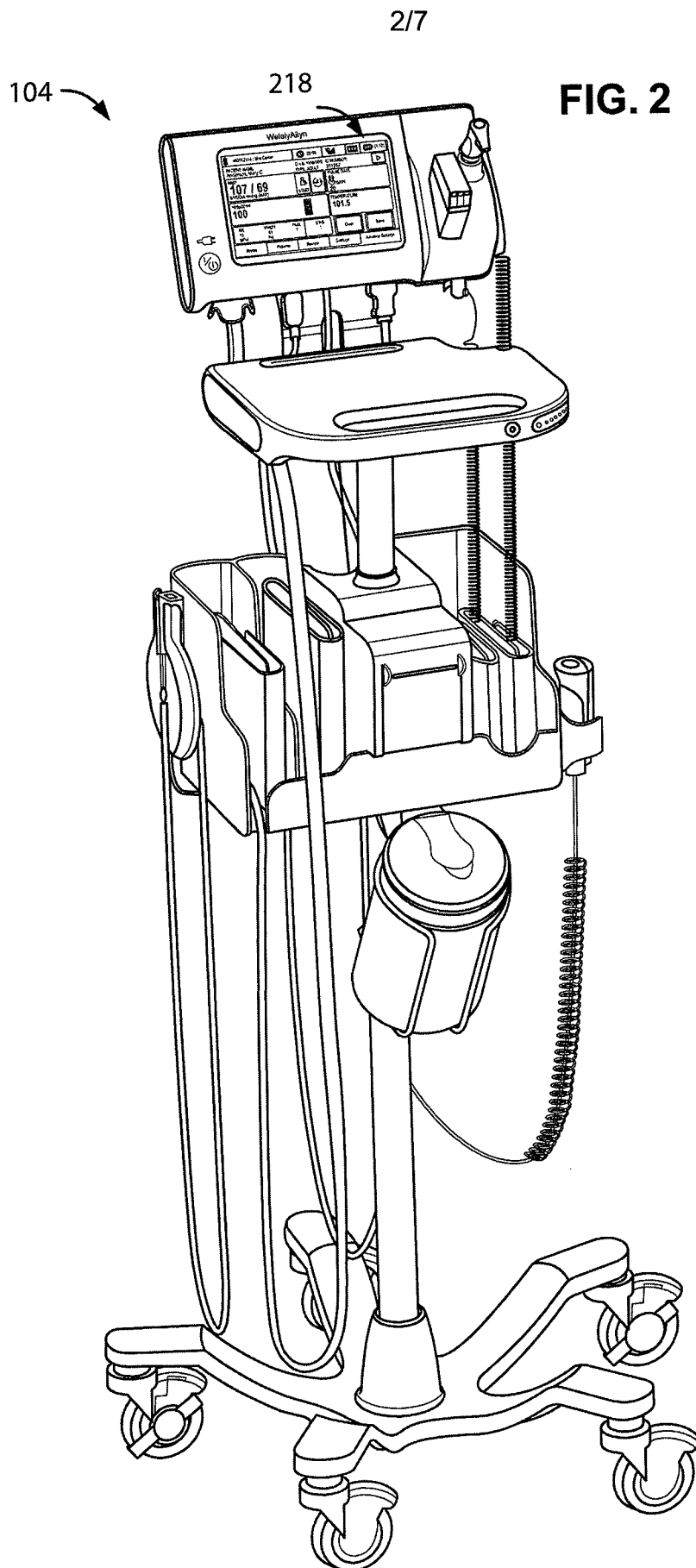
wherein one of the plurality of reusable assets is a configuration manager configured to maintain persistent configuration of the device and provide a device configuration to client software components; and

wherein one of the plurality of reusable assets is a workflow manager configured to add, remove, start or stop manager scripts in the device.

20. The medical device software system of claim 19, wherein at least one of the plurality of reusable assets is configured to enable communication between the medical device and a central unit.

1/7

**FIG. 1**



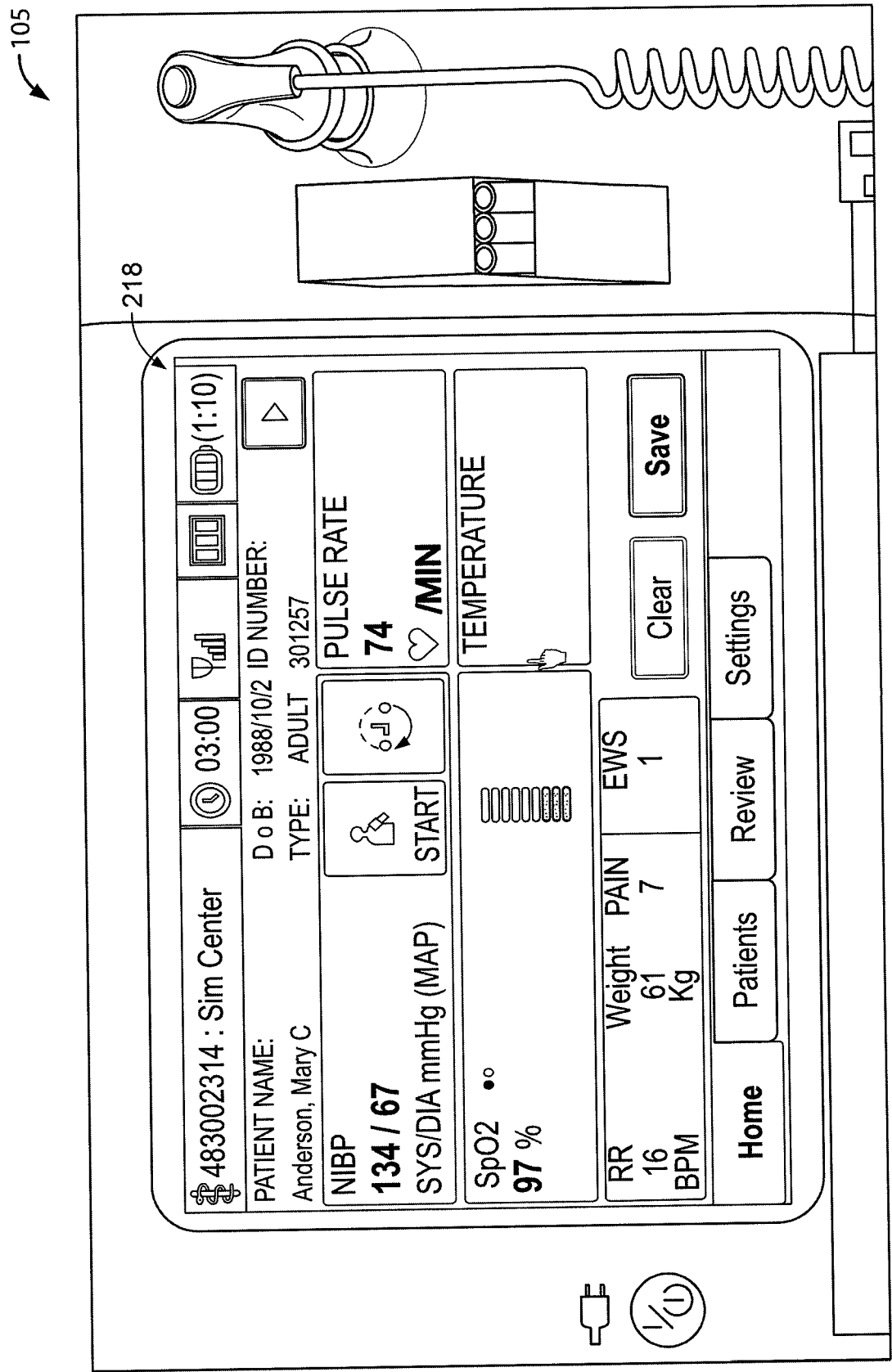
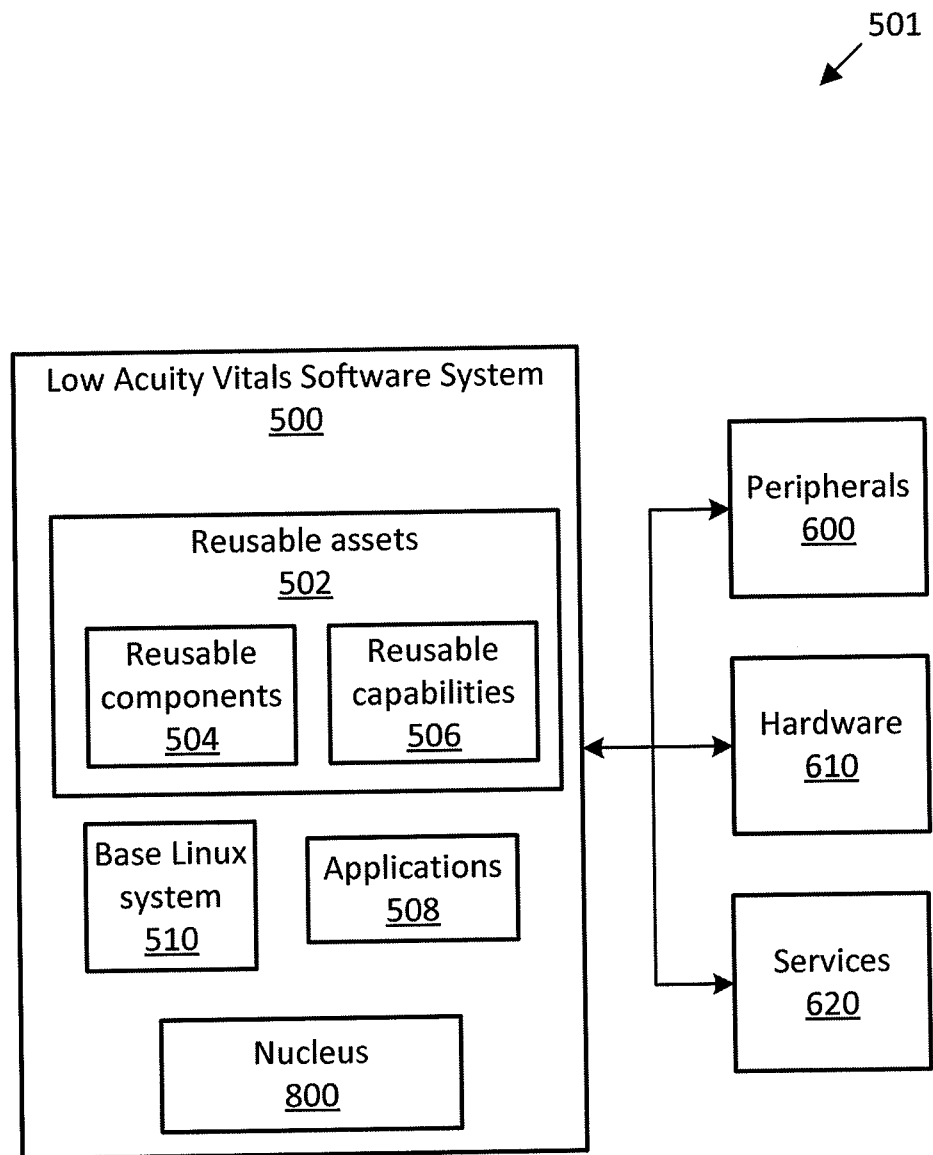
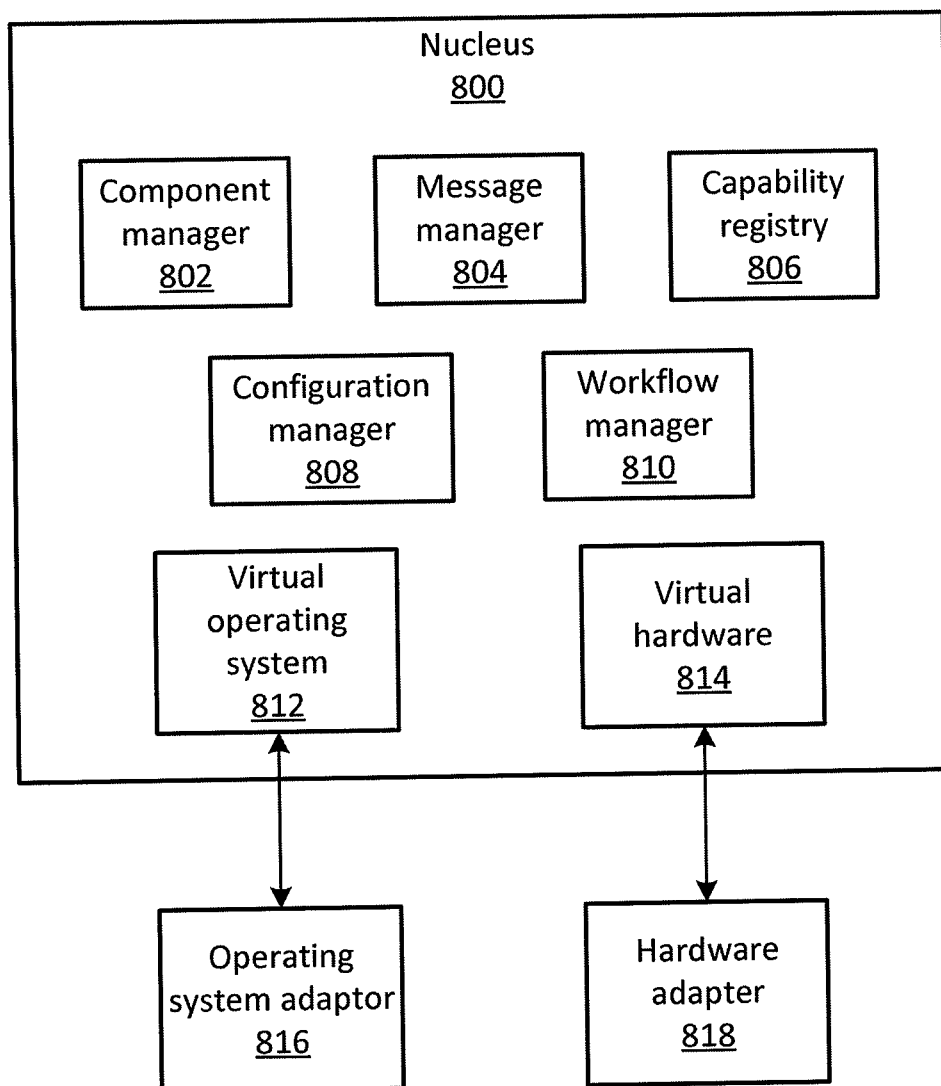


FIG. 3

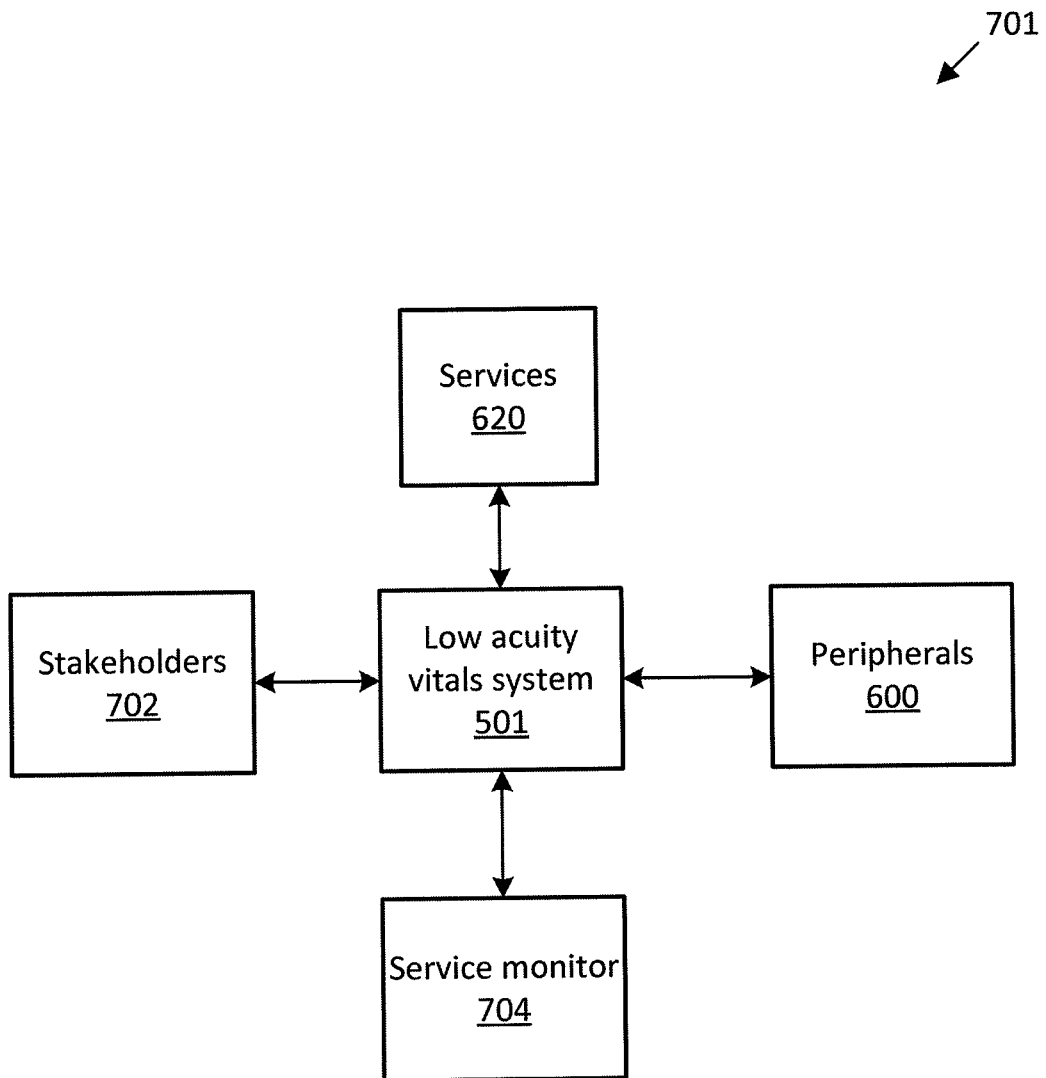
4/7

**FIG. 4**

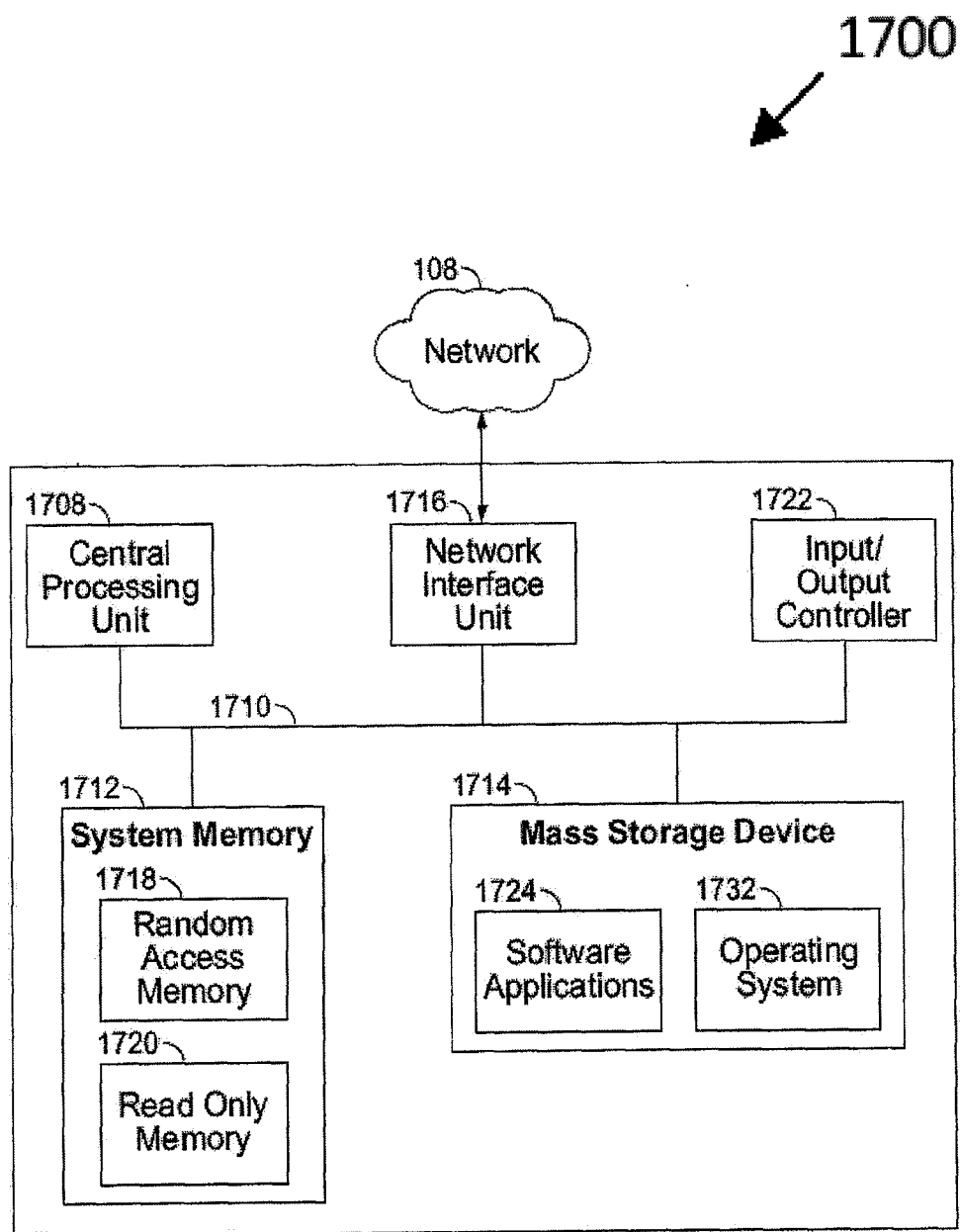
5/7

**FIG. 5**

6/7

**FIG. 6**

7/7

**FIG. 7**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/037173**A. CLASSIFICATION OF SUBJECT MATTER****A61B 5/00(2006.01)i, G06F 9/44(2006.01)i, A61B 5/01(2006.01)i, A61B 5/02(2006.01)i, A61B 5/145(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

A61B 5/00; G06F 15/16; A61N 1/362; G06F 9/44; A61B 5/024; A61B 10/02; G06F 11/00; A61B 5/01; A61B 5/145

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: medical, modular, software, reusable assets, monitoring system

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2009-0172640 A1 (ERIC P. GEISMAR et al.) 02 July 2009	1-4, 9-11, 16, 17
A	See abstract, paragraphs [0017]-[0029], claim 1 and figures 1-3.	5-8, 12-15, 18-20
Y	US 2008-0114852 A1 (MICHAEL J. CLAUS et al.) 15 May 2008	1-4, 9-11, 16, 17
A	See abstract, paragraphs [0028]-[0048], claims 1-3 and figures 1-5.	
A	US 2008-0155494 A1 (MICHAEL GOBEL) 26 June 2008	1-20
A	See abstract, paragraphs [0055]-[0073] and figures 1-6.	
A	US 2014-0155771 A1 (SENORX, INC.) 05 June 2014	1-20
A	See abstract, paragraphs [0028]-[0063] and figures 1A-7.	
A	US 2005-0159787 A1 (KURT R. LINBERG et al.) 21 July 2005	1-20
A	See abstract, paragraphs [0064]-[0081] and figures 2-6.	



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

25 September 2015 (25.09.2015)

Date of mailing of the international search report

30 September 2015 (30.09.2015)

Name and mailing address of the ISA/KR

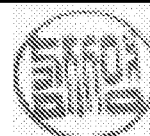
International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

KIM, Ja Young

Telephone No. +82-42-481-8131



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/037173

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009-0172640 A1	02/07/2009	WO 2009-082553 A1	02/07/2009
US 2008-0114852 A1	15/05/2008	US 7730362 B2	01/06/2010
US 2008-0155494 A1	26/06/2008	US 8650539 B2	11/02/2014
US 2014-0155771 A1	05/06/2014	AU 2004-244989 A1	16/12/2004
		AU 2004-244989 B2	26/05/2011
		CA 2527233 A1	16/12/2004
		CA 2527233 C	02/08/2011
		EP 1628580 A1	01/03/2006
		EP 1628580 B1	24/07/2013
		ES 2428357 T3	07/11/2013
		JP 2007-525246 A	06/09/2007
		US 2005-004559 A1	06/01/2005
		US 2009-030405 A1	29/01/2009
		US 2010-114335 A1	06/05/2010
		US 8652121 B2	18/02/2014
		US 8696650 B2	15/04/2014
		WO 2004-107989 A1	16/12/2004
US 2005-0159787 A1	21/07/2005	CA 2476280 A1	14/08/2003
		DE 60037035 D1	20/12/2007
		DE 60037035 T2	28/08/2008
		EP 1242143 A1	25/09/2002
		EP 1242143 B1	07/11/2007
		EP 1478432 A2	24/11/2004
		EP 1478432 B1	15/07/2009
		JP 2005-518006 A	16/06/2005
		JP 4364644 B2	18/11/2009
		US 2002-0123673 A1	05/09/2002
		US 2003-0041866 A1	06/03/2003
		US 2006-0189854 A1	24/08/2006
		US 6497655 B1	24/12/2002
		US 6878112 B2	12/04/2005
		US 7060031 B2	13/06/2006
		US 7815568 B2	19/10/2010
		WO 01-43823 A1	21/06/2001
		WO 03-066159 A2	14/08/2003
		WO 2007-103844 A1	13/09/2007

专利名称(译)	低视力生命体系结构		
公开(公告)号	EP3160332A1	公开(公告)日	2017-05-03
申请号	EP2015811743	申请日	2015-06-23
[标]申请(专利权)人(译)	伟伦公司		
申请(专利权)人(译)	伟伦，INC.		
当前申请(专利权)人(译)	伟伦，INC.		
[标]发明人	GONDEK CORY R NAGASUBRAMANIAN SOUNDHARYA KOVITZ PAUL S ALFRED CHARLES GOLDMAN DAN E		
发明人	GONDEK, CORY R. NAGASUBRAMANIAN, SOUNDHARYA KOVITZ, PAUL S. ALFRED, CHARLES GOLDMAN, DAN E.		
IPC分类号	A61B5/00 G06F9/44 A61B5/01 A61B5/02 A61B5/145		
CPC分类号	A61B5/01 A61B5/021 A61B5/024 A61B5/14542 G06F8/70 G06F9/44505 G06F9/542 G06F9/546 G16H40/40		
优先权	62/017895 2014-06-27 US 62/017908 2014-06-27 US 14/535395 2014-11-07 US 14/535401 2014-11-07 US		
其他公开文献	EP3160332A4		
外部链接	Espacenet		

摘要(译)

公开了一种医疗设备，其包括存储被编程为提供医疗设备功能的多个模块化软件组件的非暂时性存储设备。软件组件可以配置为独立运行作为守护程序，无需重新编译未更改的软件组件即可进行修改，并通过软件组件相互通信。还公开了一种包括被配置为与医疗设备通信的中央单元的系统。还公开了一种存储在非暂时性计算机可读存储介质上的医疗设备软件架构，该存储介质包括被编程以提供医疗设备功能的多个模块化软件组件。